Analyzing the Optimal Neighborhood: Algorithms for Budgeted and Partial Connected Dominating Set Problems

Samir Khuller Manish Purohit

Kanthi Sarpatwar

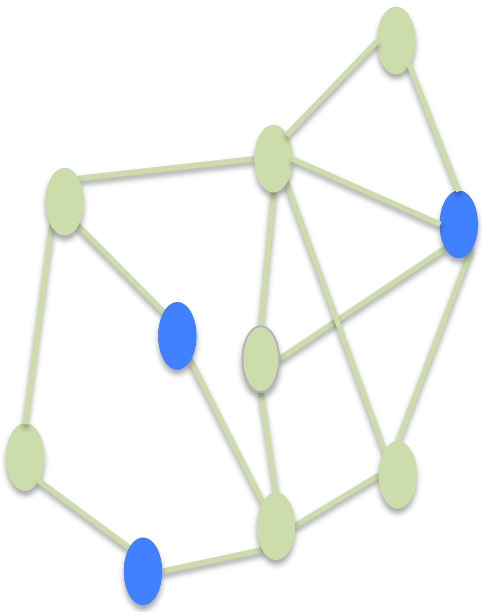
Department of Computer Science

University of Maryland, College Park

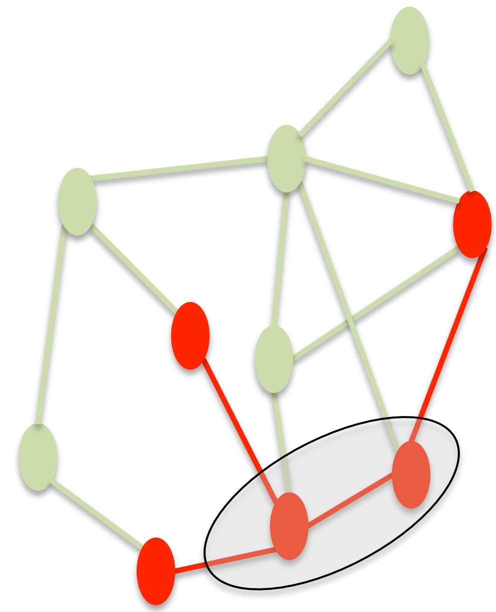
Outline

- **Problems**
  - **Connected dominating set**
  - Generalizations and variants
- Our Work
  - Summary
  - Algorithm
  - Analysis
- Future Work

Connected Dominating Set (CDS)



Dominating Set

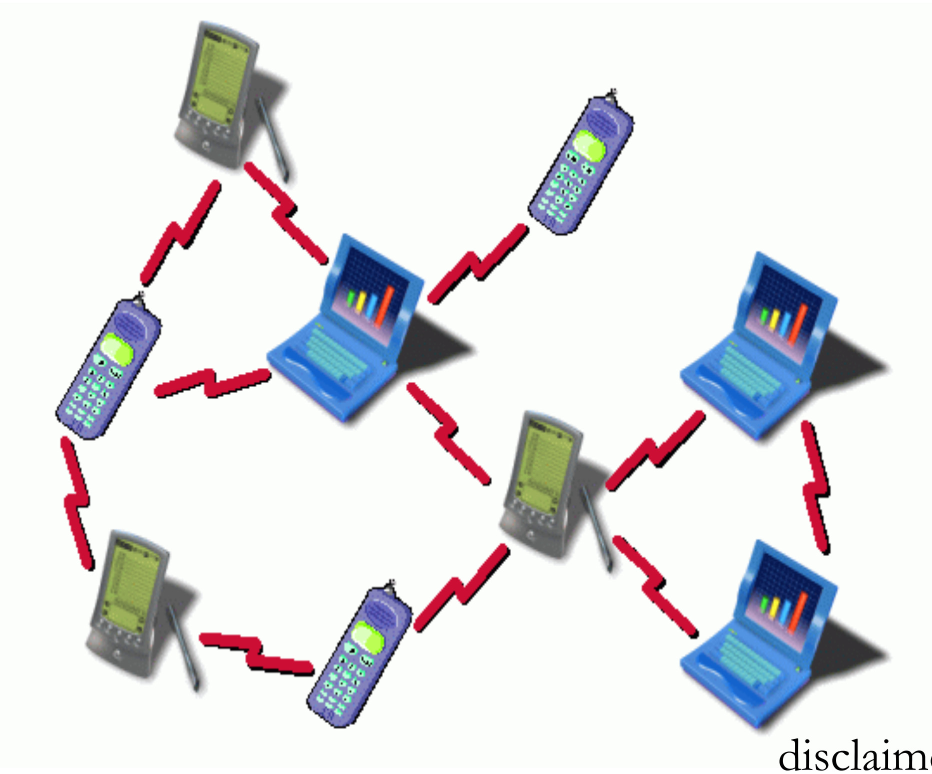


Connected Dominating set

Input: Graph, G = (V,E)

Output: Min cost dominating set that induces a connected sub-graph.

Motivation for CDS



disclaimer: from Internet

Good basic model for virtual backbone in ad hoc networks [BD 97].

Approximation Results for CDS

Arbitrary Graphs (max degree Δ) : ln Δ + 3 [GK 97]

Distributed Setting : O(log n) [DMPRS 97]

PTAS in Special Graphs
- Planar Graphs [DH 05]
- Geometric Graphs [CHLWD 03]

On general graphs, it is set cover hard. We cannot hope for better than $O(\log^{1-\varepsilon} \Delta)$.

Outline

- **Problems**
  - Connected dominating set
  - **Generalizations and variants**
- Our Work
  - Summary
  - Algorithm
  - Analysis
- Future Work

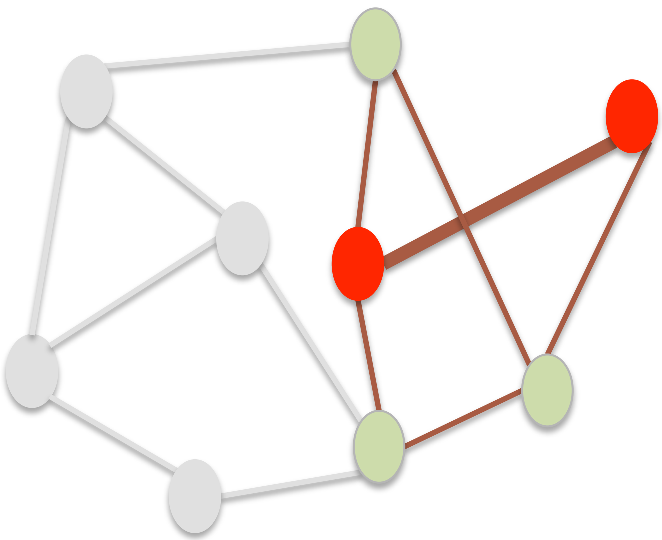
Partial CDS (PCDS)

Input:

- Graph, G=(V,E)
- Quota, Q

Output:

Find a min cost connected sub-graph that covers at least Q vertices



Quota Q = 5

Connected sub-graph that covers ~~all~~ **a given fraction of the vertices** – ~~CDS~~ **Partial CDS (PCDS)**

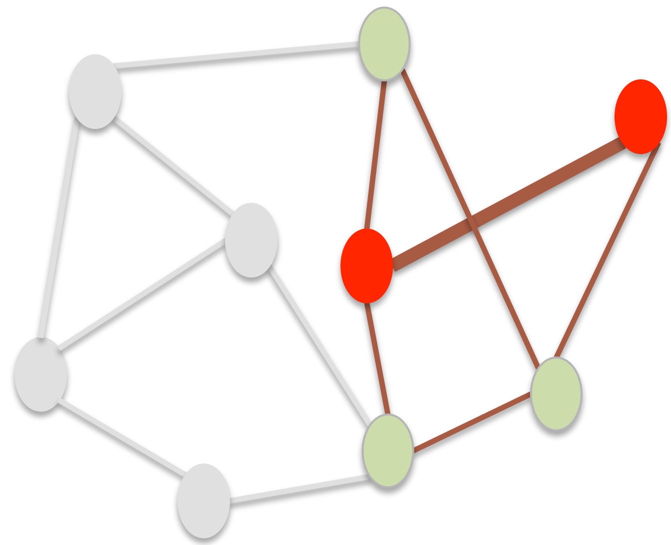
Budgeted CDS (BCDS)

Input:

- Graph, G = (V,E)
- Budget, k

Output:

Find a connected sub-graph on k vertices that dominates a maximum number of vertices



Budget k = 2
Coverage = 5

Dual problem of PCDS – Budgeted CDS (BCDS)

Prior Work on PCDS and BCDS

No prior non-trivial approximation known for either of the problems.

Heuristics based study done in sensor networks[LL 05].

Problems also studied in a "local information setting"[ABNRT 13, KL 12].

Related Work

Partial and Budgeted variants of several optimization problems have been studied in literature

minimum spanning tree [BRV 96, Garg 97-05, AK 00]

Steiner tree [CRW 06]

Steiner forest [HJ 06, SS 06, GHNR 08, AK 08]

k-center, k-median, facility location [CKMN 01]

vertex cover [HS 02, Mestre 09]

Outline

- Problems
  - Connected dominating set
  - Generalizations and variants
- **Our Work**
  - **Summary**
  - Algorithm
  - Analysis
- Future Work

Our Results

A polynomial time algorithm for the PCDS problem with an approximation guarantee **$4 \ln \Delta + 2$**.

A polynomial time algorithm for the BCDS problem with an approximation guarantee **$(1/13)(1-1/e)$**.

Further generalizations

Special submodular function: We call a function (with domain as the vertex set of a graph) special submodular if it is submodular and has the property that for any subsets A and B of the domain, such that the vertices of A and B do not share neighbors, then f(A U B) = f(A) + f(B).

A polynomial time algorithm for the special submodular PCDS problem with an approximation guarantee **4 ln Q + 2**.

A polynomial time algorithm for the special submodular BCDS problem with an approximation guarantee **(1/13)(1-1/e)**.

Outline

- Problems
  - Connected dominating set
  - Generalizations and variants
- **Our Work**
  - **Summary**
  - **Algorithm**
  - Analysis
- Future Work

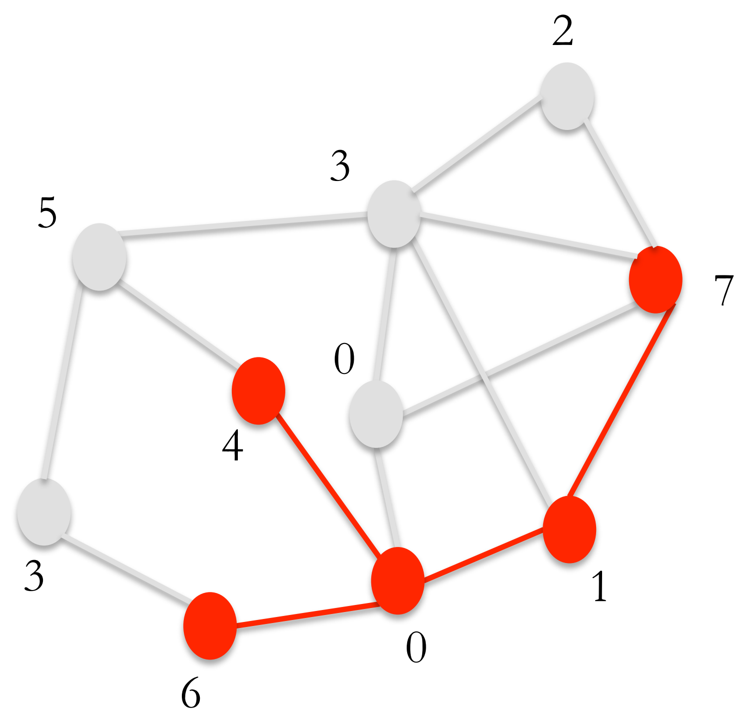
Quota Steiner Tree (QST)

Input

- Graph, G=(V,E)
- Cost function, c: E -> Z^+
- Profit function, p: V -> Z^+
- Quota, Q

Output

- Min cost tree with total profit at least Q

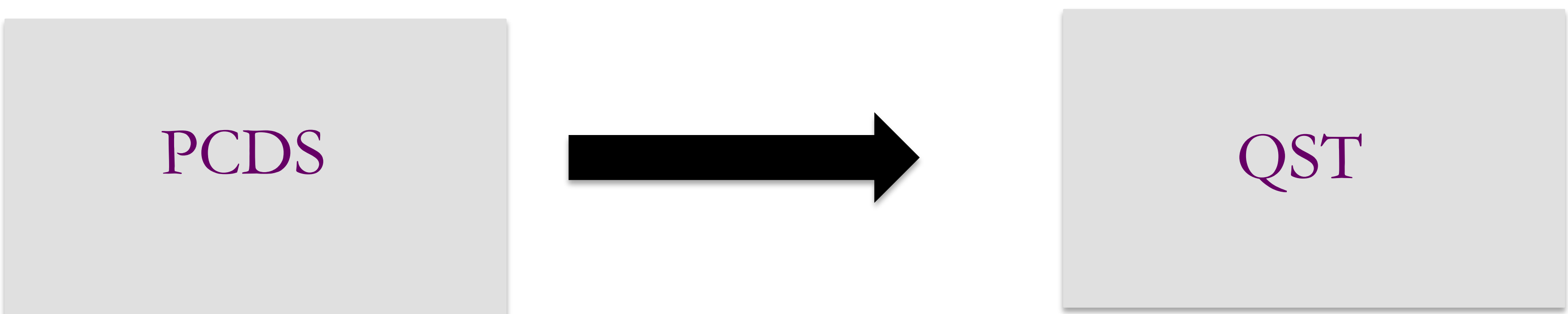


QST with Q = 18

There is a 2-approximation for QST [JMP 00, Garg 05]

Our Algorithm: Main Idea

We obtain a reduction from PCDS to QST with a $O(\log \Delta)$ factor loss in approximation factor



Non-linear but submodular profit function

Linear profit function

Approximate the submodular function by a linear profit function

Candidates for approximate linear functions

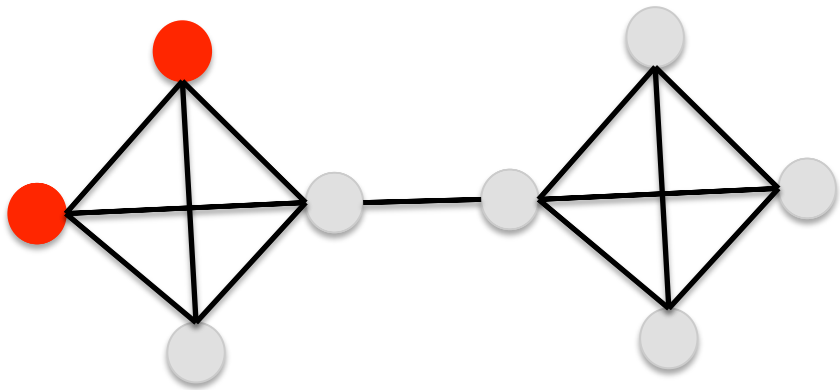
How about the degree function, $p(v) = d(v) + 1$?

Candidates for approximate linear functions

How about the degree function, $p(v) = d(v) + 1$?

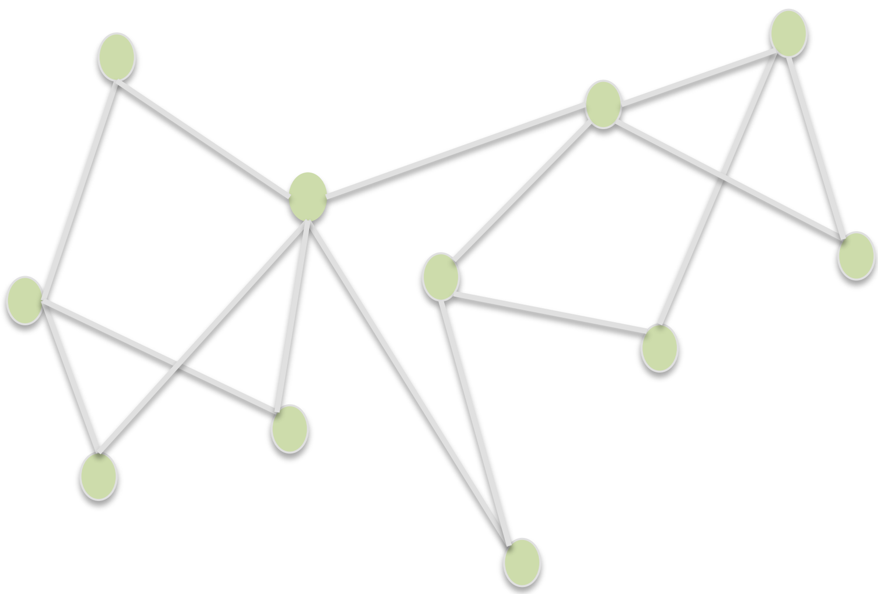
Not a good idea !!

Quota = 8



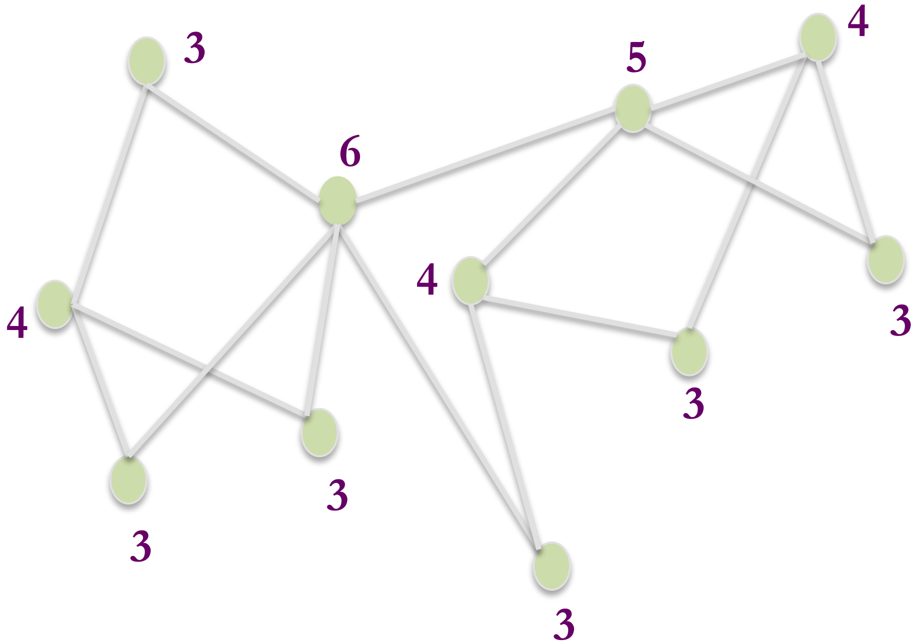
Red nodes give a profit of 8 according to the linear degree function. But the coverage is only 4.

Candidates for approximate linear functions



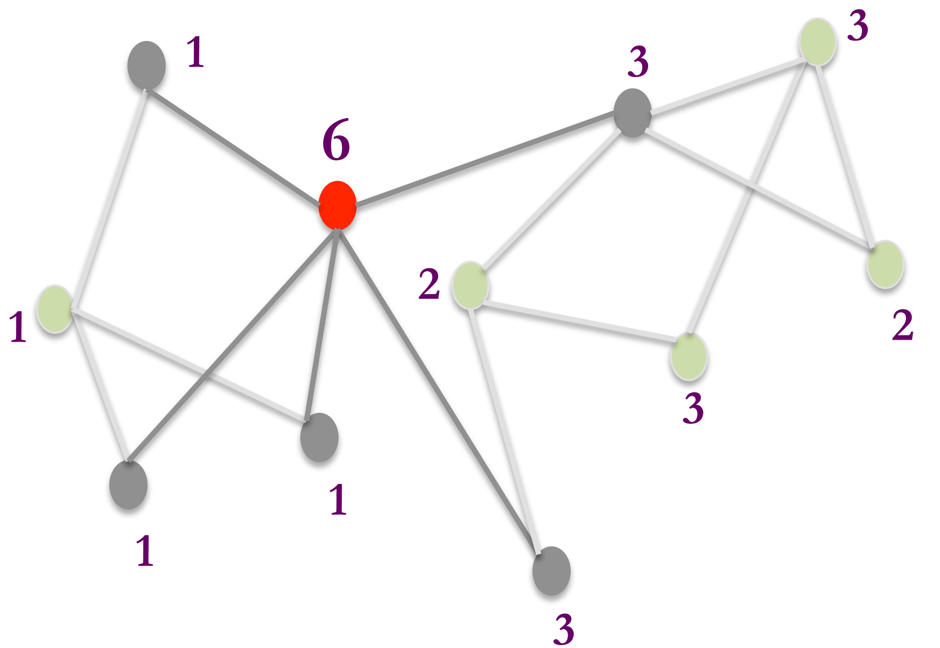
Using the greedy algorithm for dominating set to define the linear function

Candidates for approximate linear functions



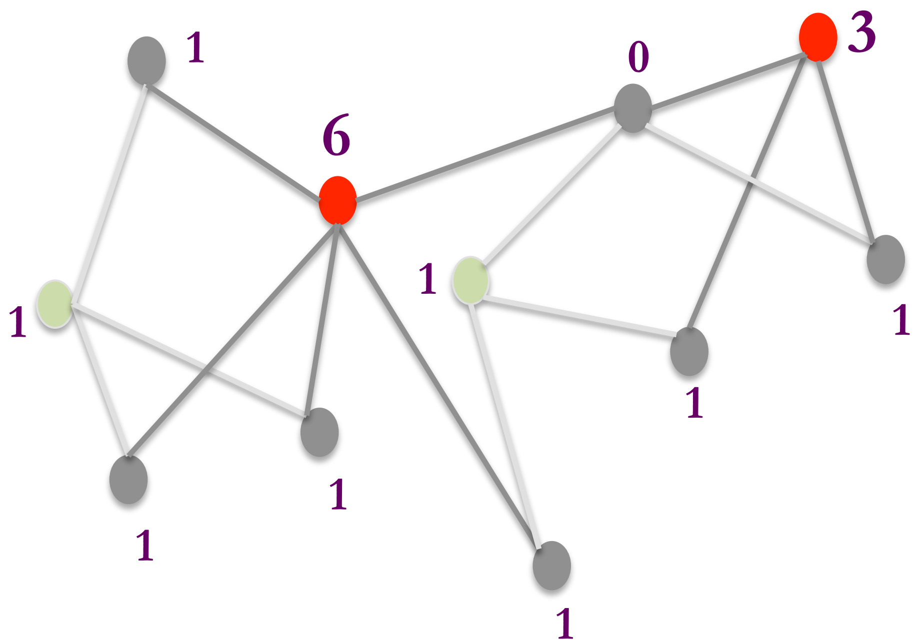
For each vertex, compute the number of vertices that will be covered for the first time.

Candidates for approximate linear functions



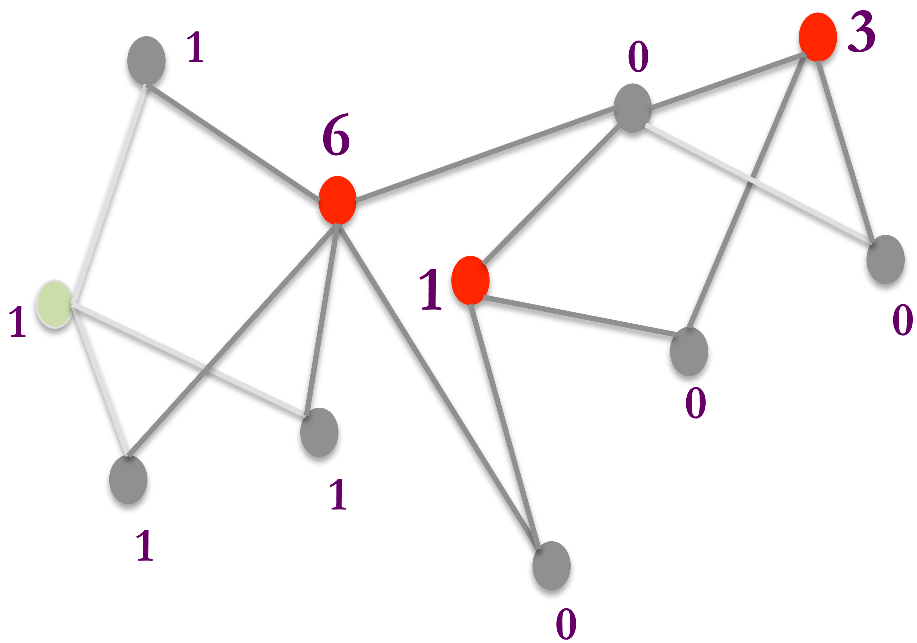
We choose the vertex with maximum profit. Re-compute the new profit function.

Candidates for approximate linear functions



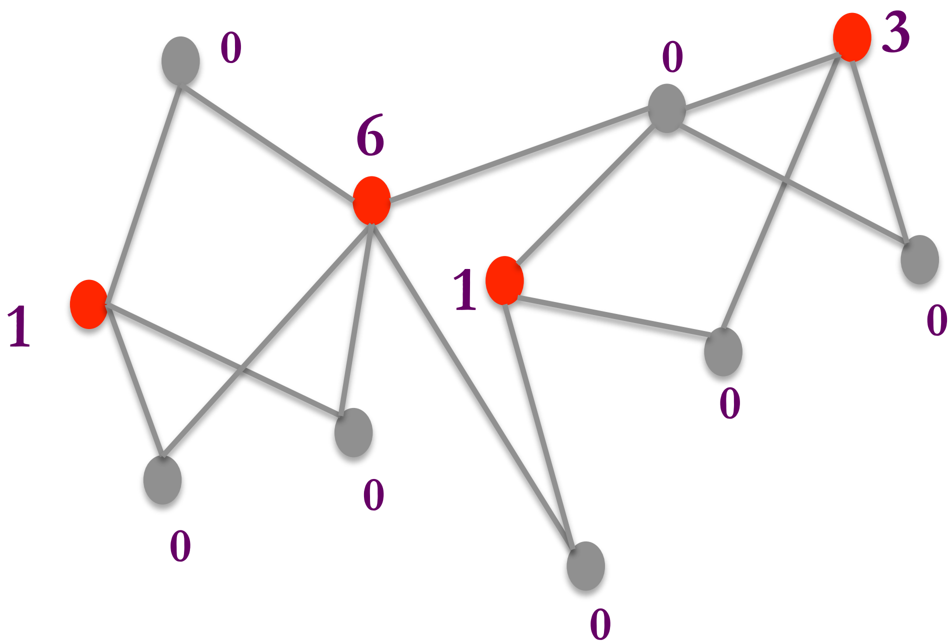
Tie breaking is arbitrary.

Candidates for approximate linear functions



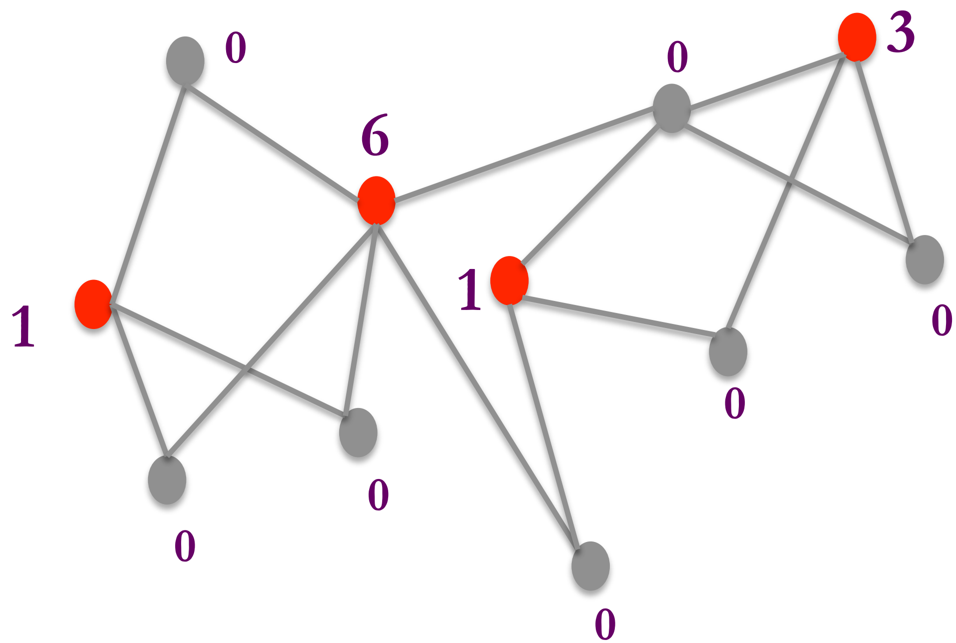
Tie breaking is arbitrary.

Candidates for approximate linear functions



This algorithm defines a natural linear profit function.

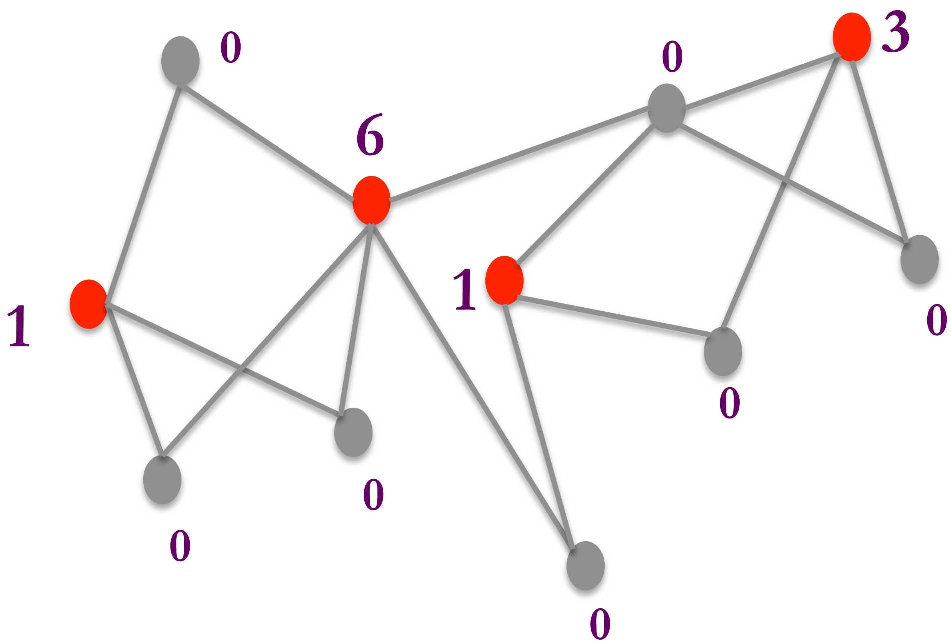
Candidates for approximate linear functions



The profit function p:V->N, is defined as

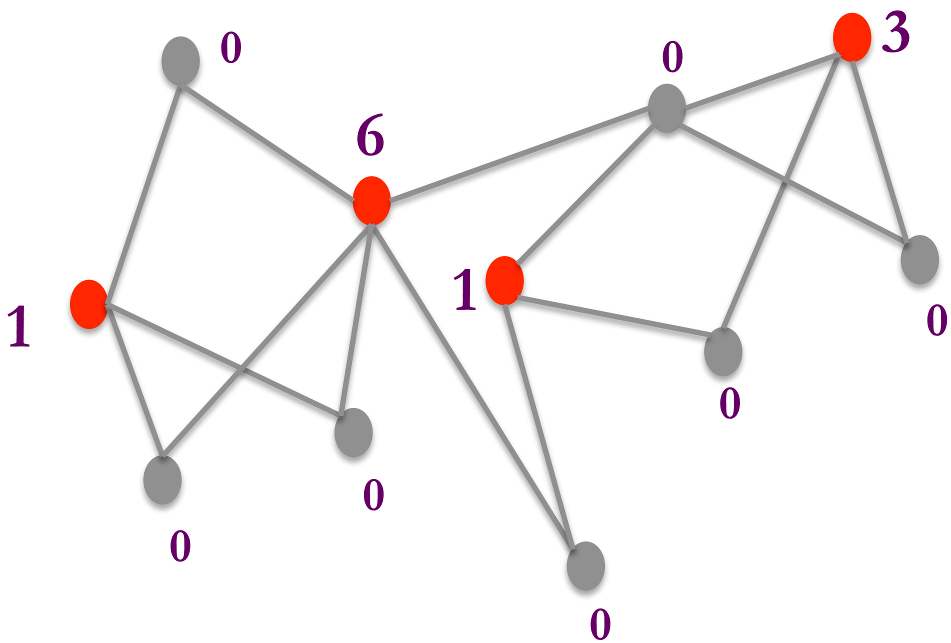
p(v) = Number of newly covered vertices when v is chosen
0, if the vertex is not chosen

Candidates for approximate linear functions



But is it a good approximation to the submodular function?

Candidates for approximate linear functions



Surprisingly …

YES !!

Our Algorithm

Step 1

Run the greedy dominating set algorithm and compute the linear profit function as the number of newly covered vertices.

Step 2

Solve the QST instance defined by this linear function.

Outline

- Problems
  - Connected dominating set
  - Generalizations and variants
- **Our Work**
  - Summary
  - Algorithm
  - **Analysis**
- Future Work

Analysis: Main Idea

OPT(PCDS) Optimal solution size for the PCDS instance.

There exists a tree T with size at most $(2 \ln \Delta + 1)$ OPT(PCDS) and $p(T) \geq Q$.

2-approx. for QST → $4\log\Delta + 2$ approx. for PCDS

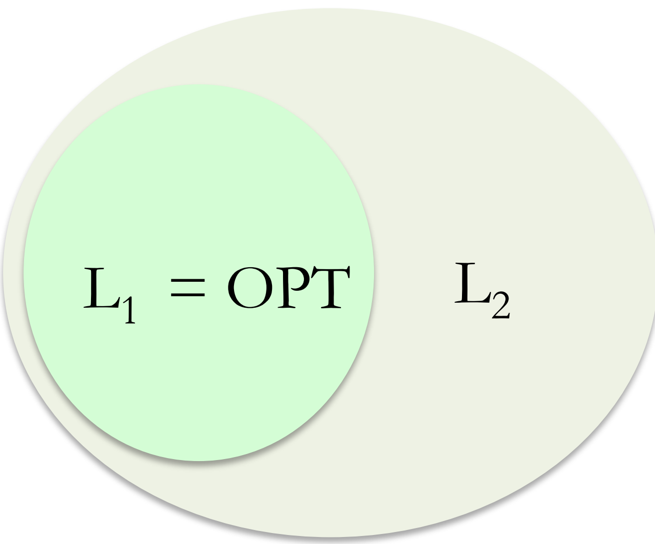
Analysis: Analyzing the Optimal Neighborhood

Optimal Solution for PCDS.

$L_1 = OPT$

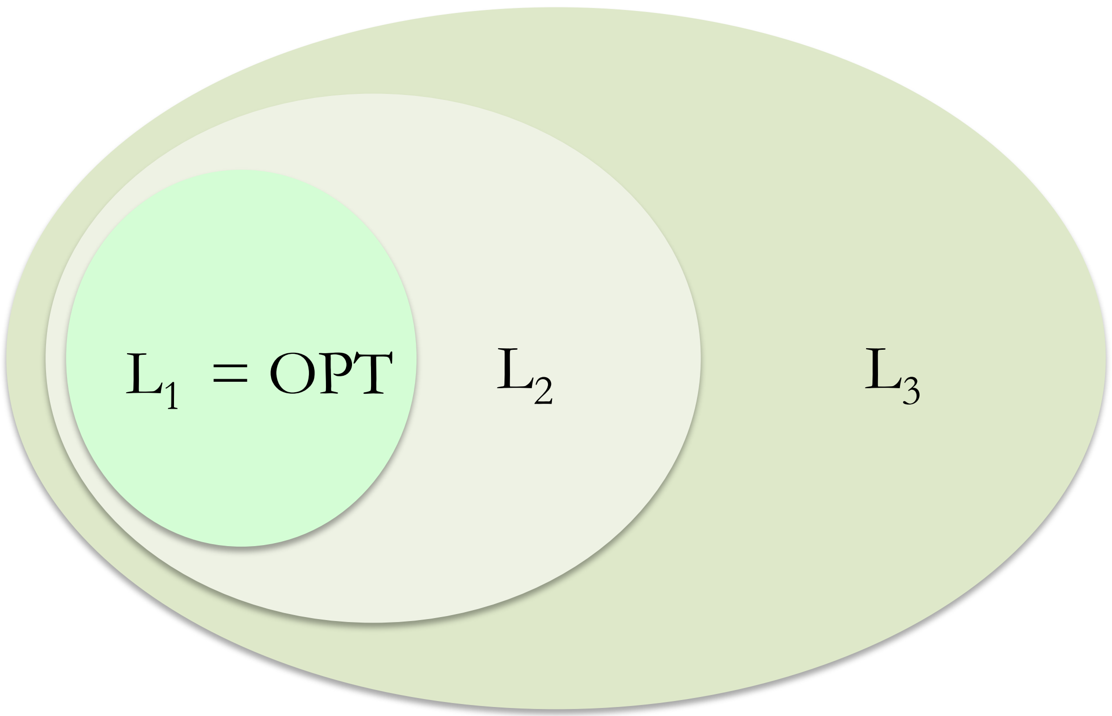
Analysis: Analyzing the Optimal Neighborhood

L_2 = Neighbors of L_1 , not in L_1



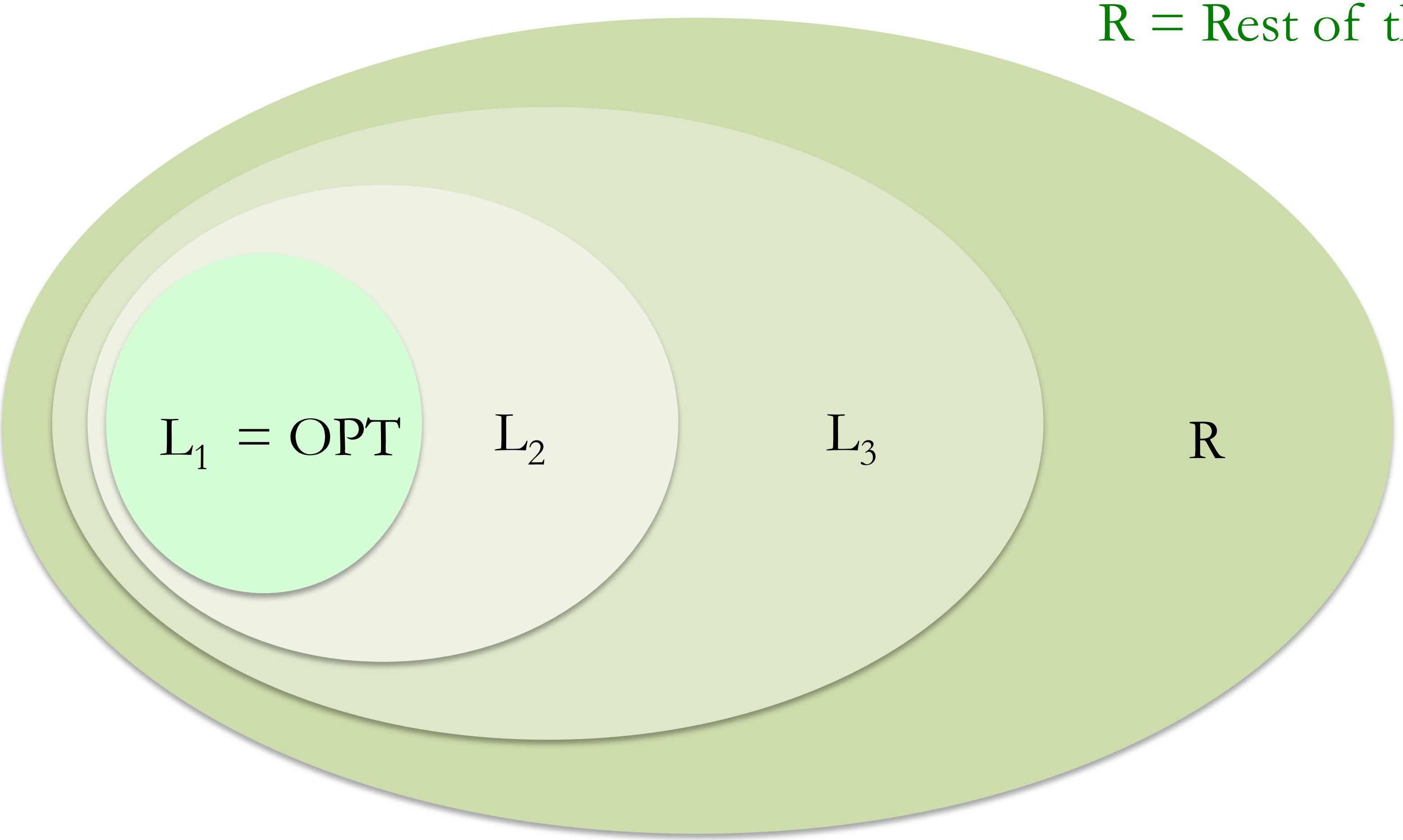
Analysis: Analyzing the Optimal Neighborhood

L_3 = Neighbors of L_2, not in L_1 or L_2



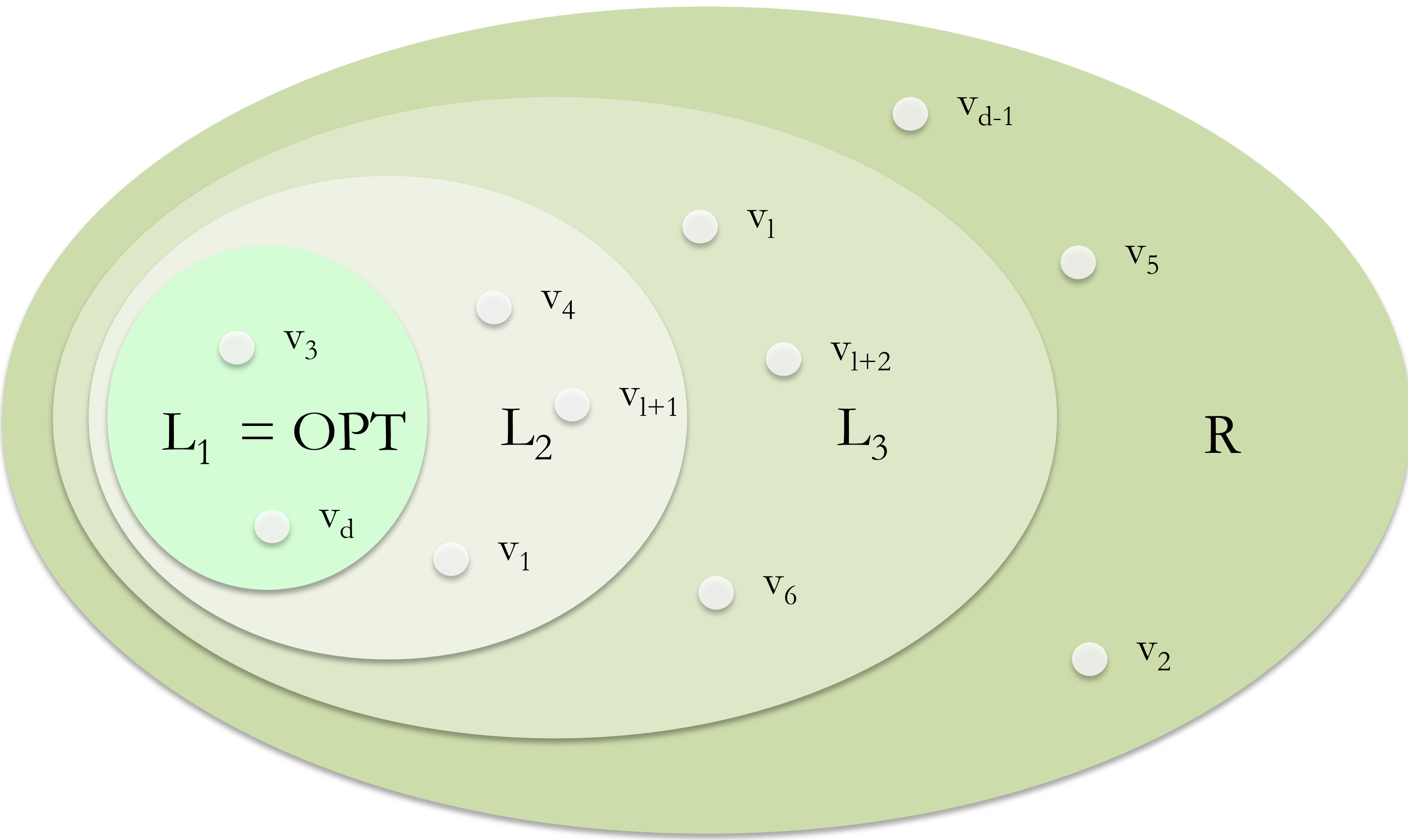
Analysis: Analyzing the Optimal Neighborhood

R = Rest of the vertices

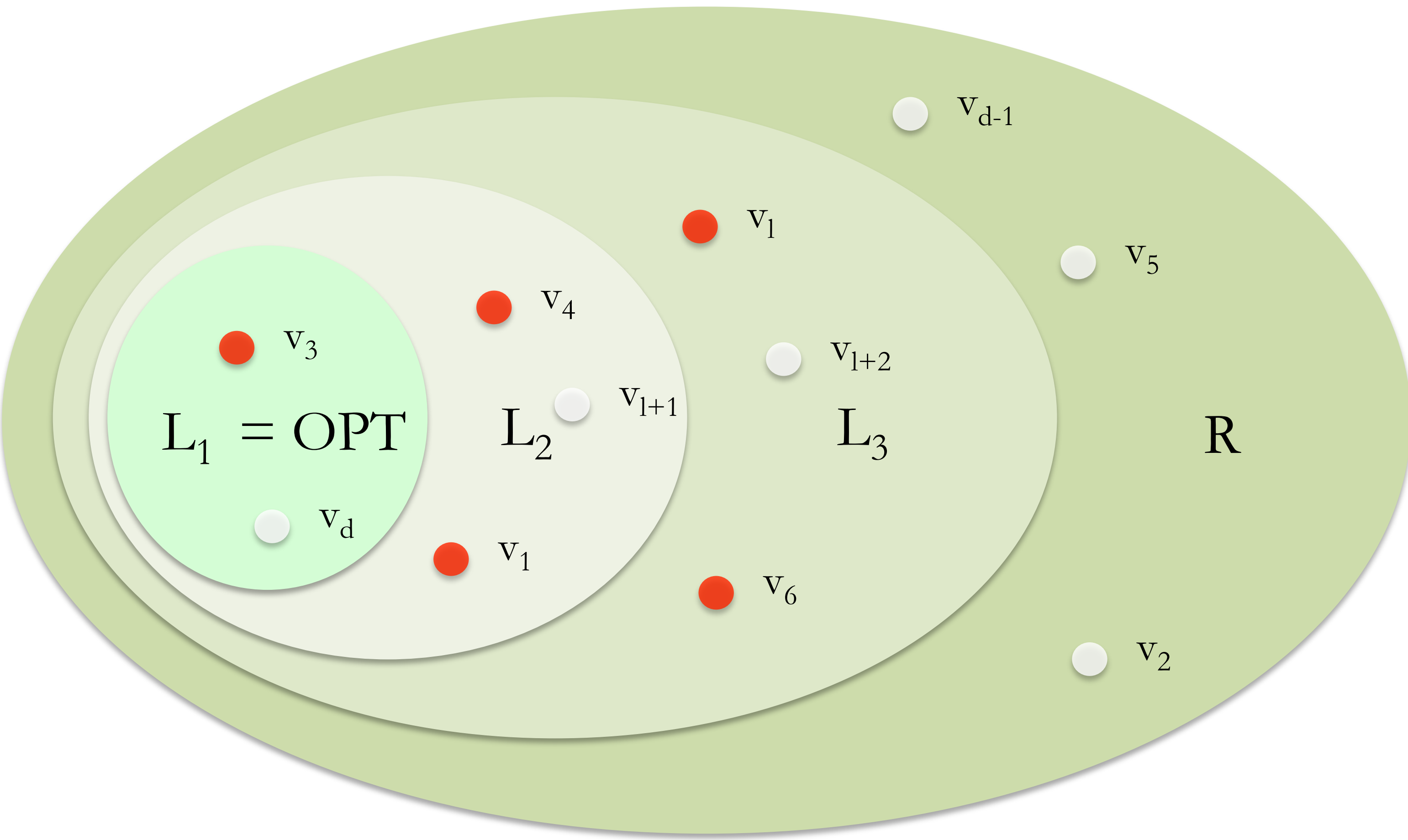


Greedy algorithm picks white vertices in the order

- $v_1, v_2, v_3, \ldots, v_d$

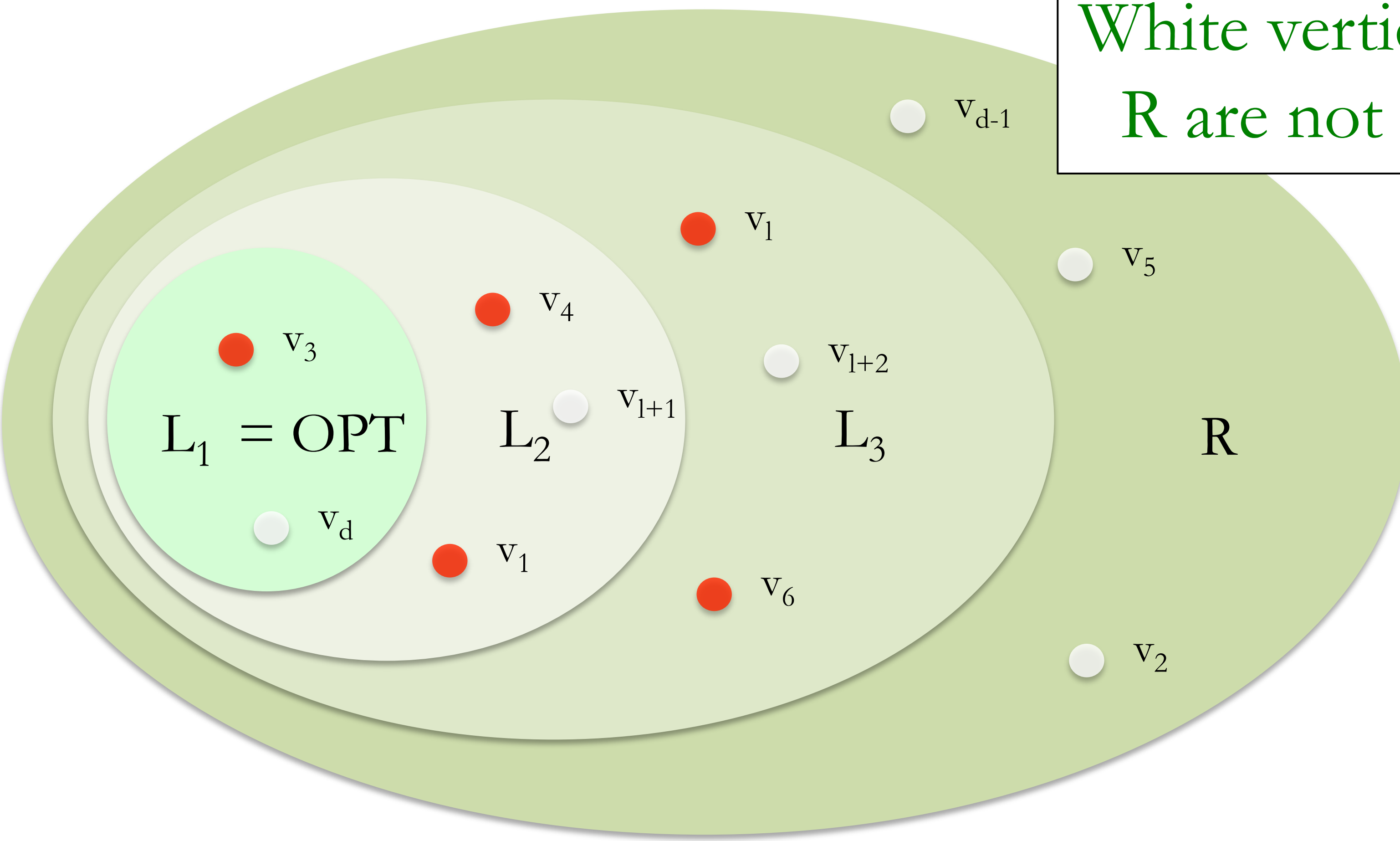


Pick vertices from L_1, L_2, L_3 in the same order as greedy until the total profit is $\geq Q$

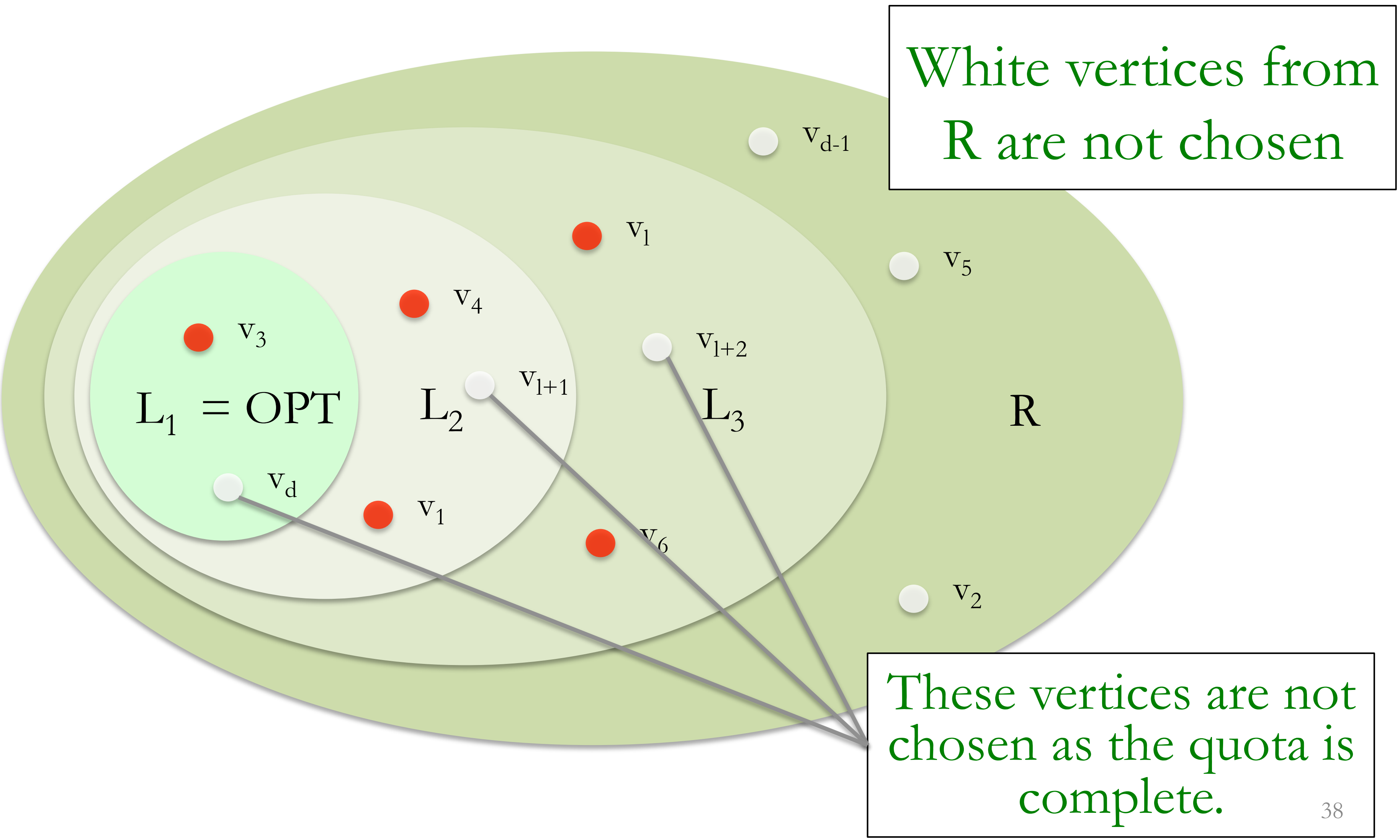


Pick vertices from L_1, L_2, L_3 in the same order as greedy until the total profit is $\geq Q$

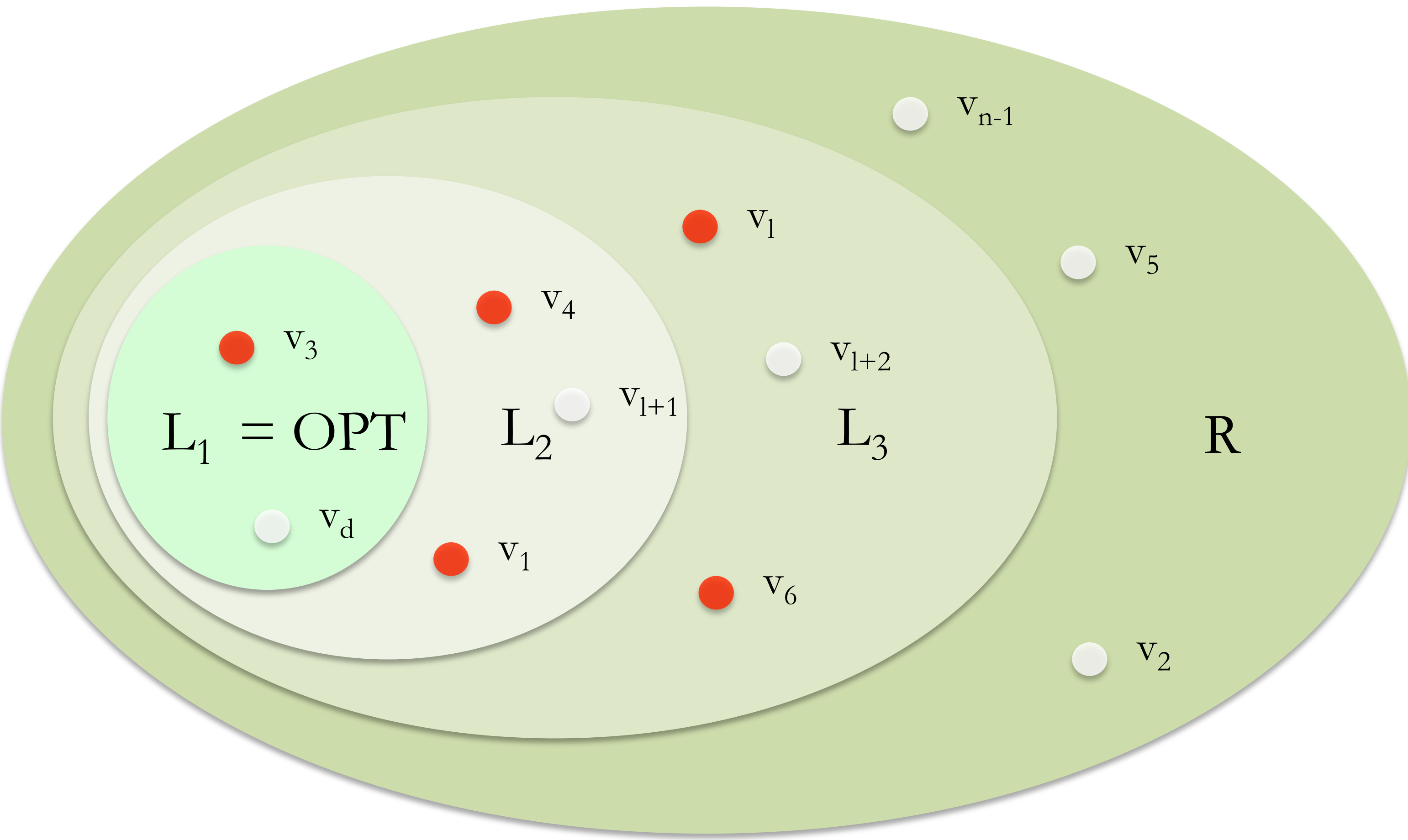
White vertices from R are not chosen



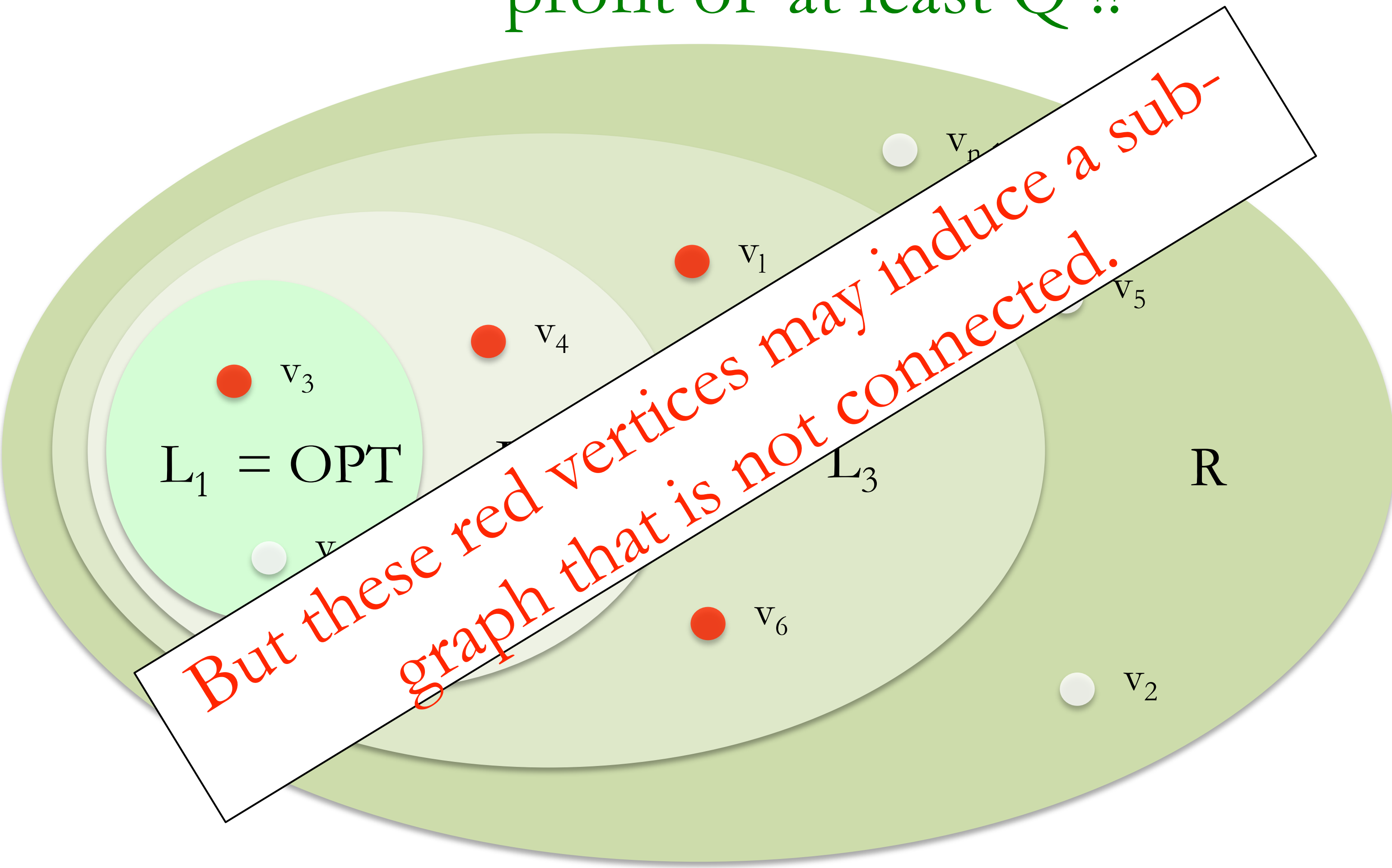
Pick vertices from L_1, L_2, L_3 in the same order as greedy until the total profit is $\geq Q$



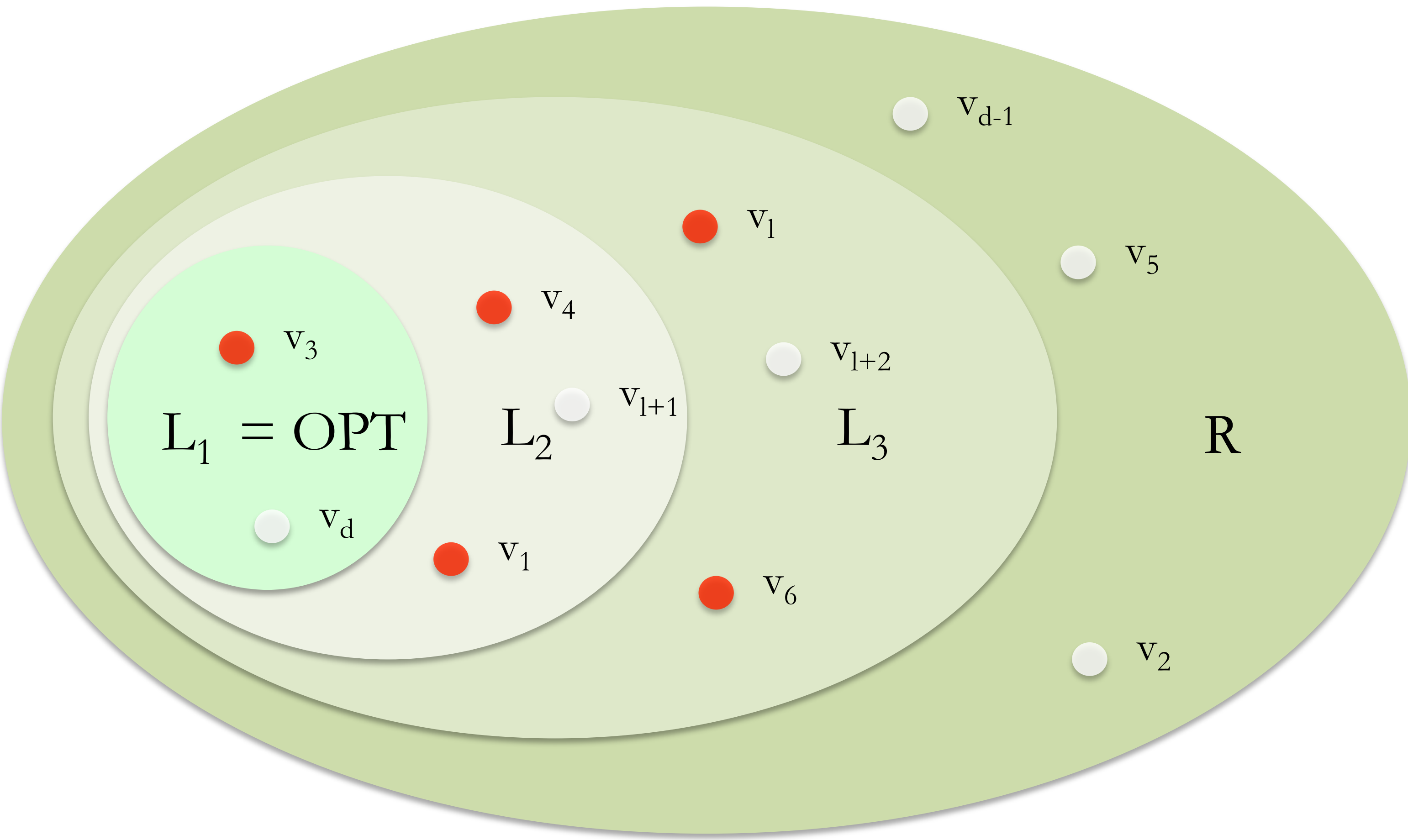
We show that number of red vertices is at most $|OPT| \ln \Delta$ and by definition they have a total profit of at least Q !!



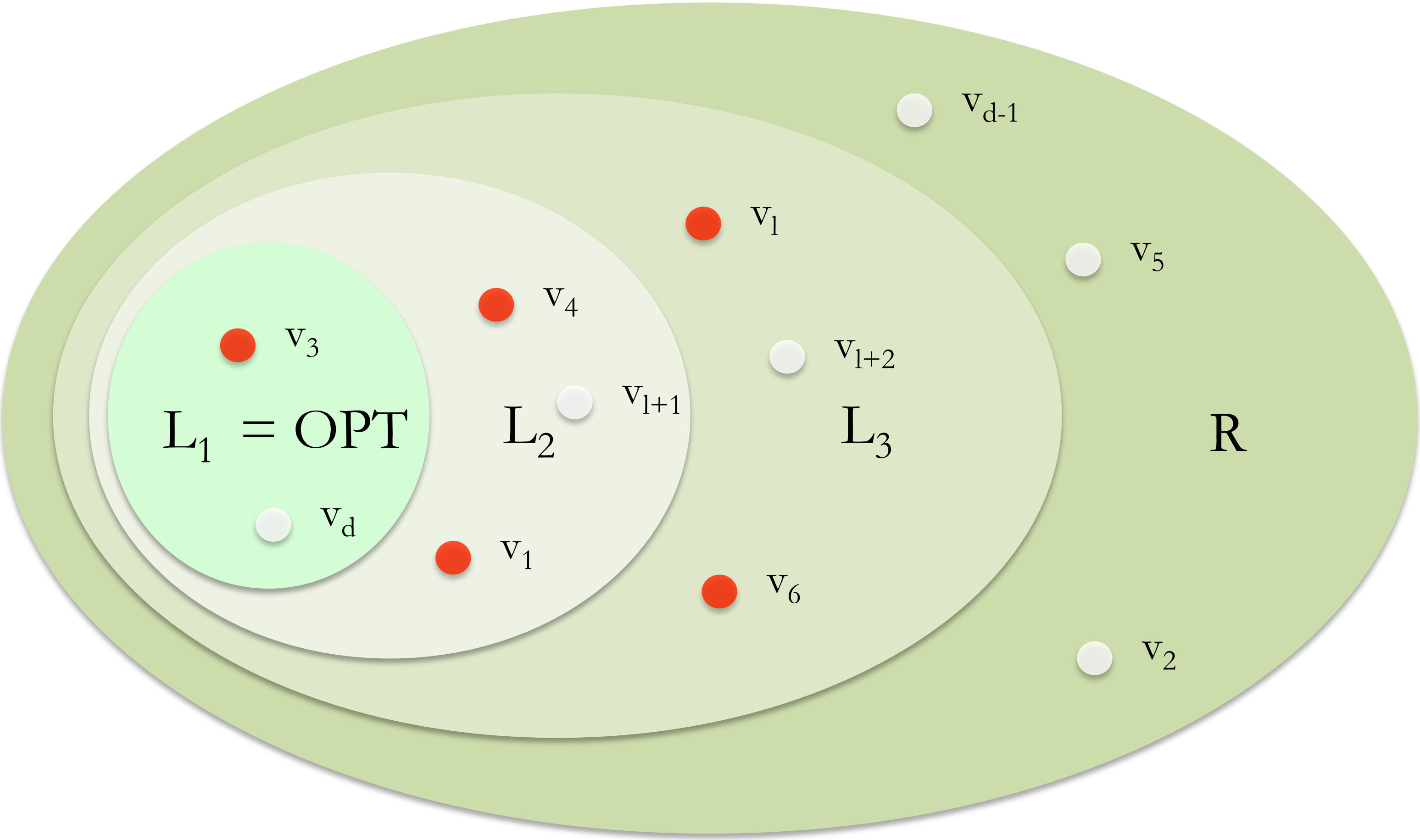
We show that number of red vertices is at most $|OPT| \ln \Delta$ and by definition they have a total profit of at least Q !!



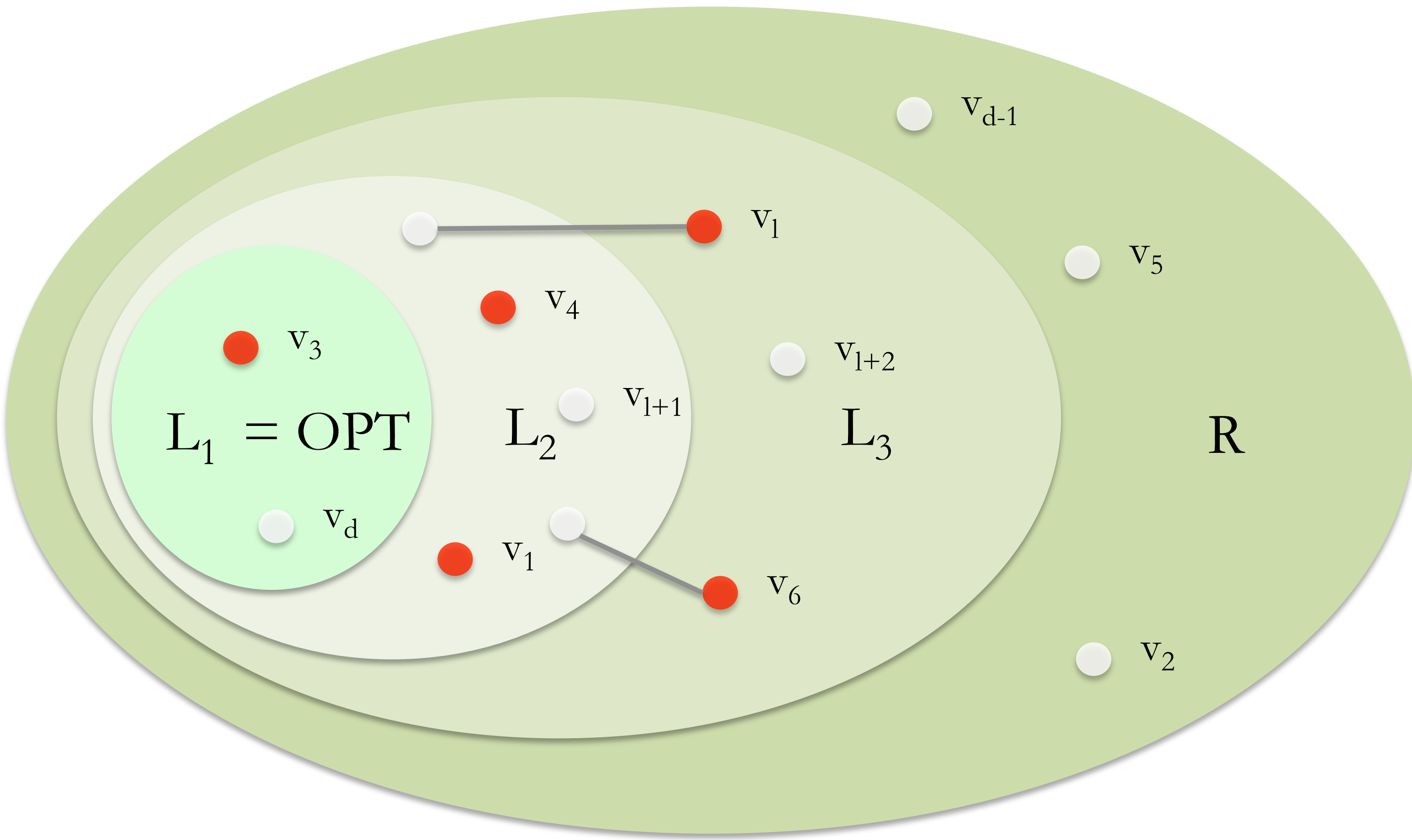
Fortunately, we can connect them easily by adding only a few more vertices.



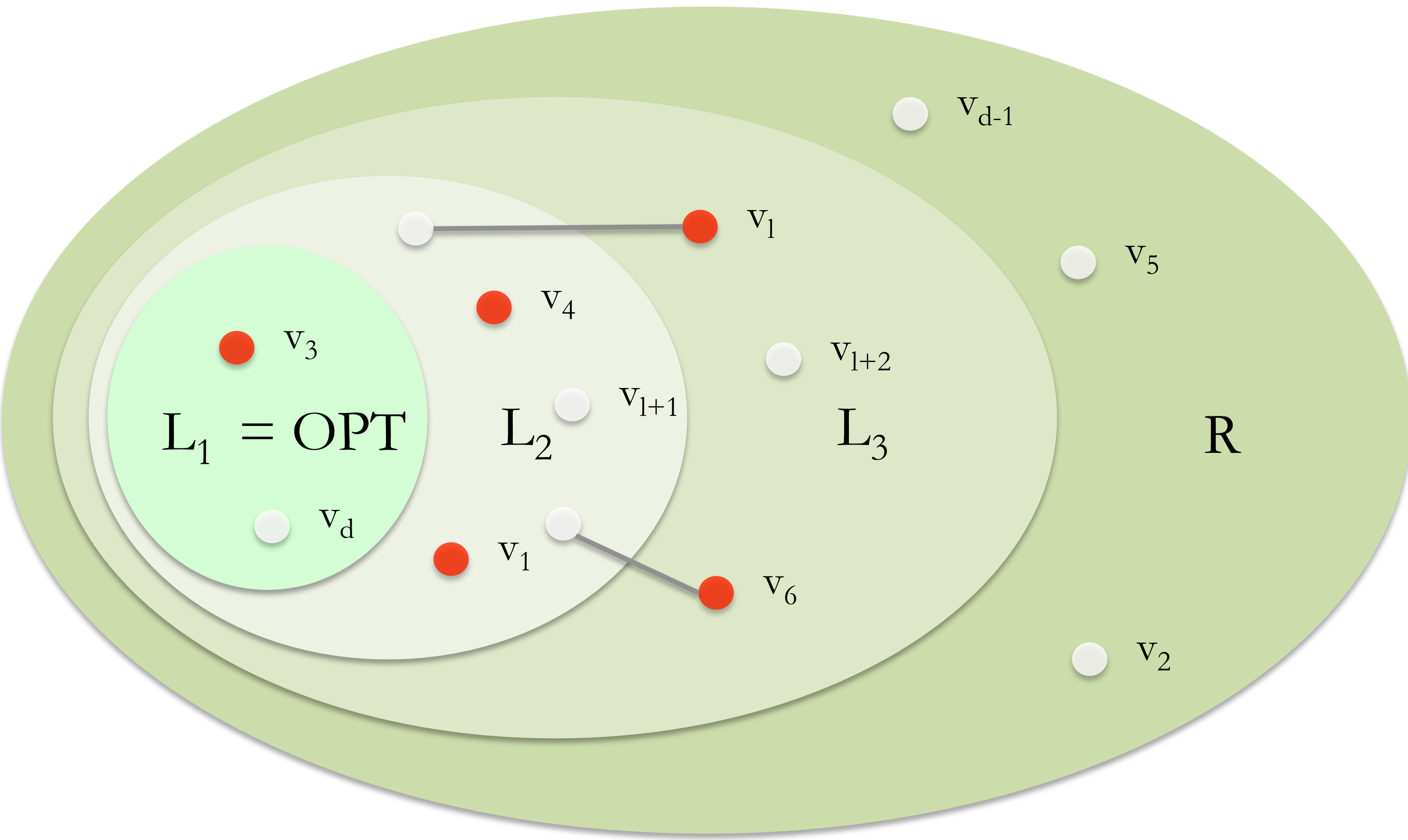
Observe that adding L_1 connects all the red vertices in L_1 and L_2.



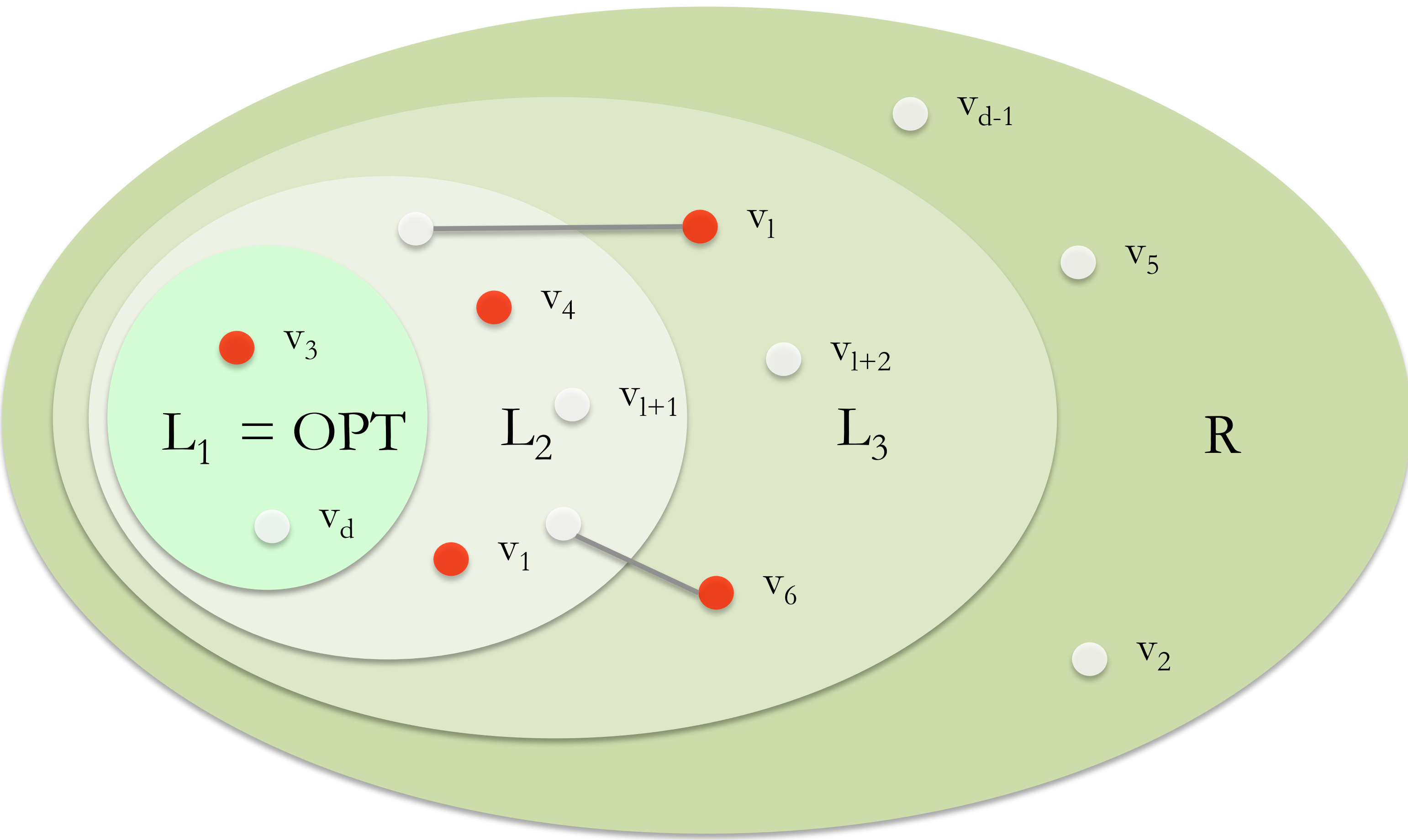
Now for every red vertex in L_3, we add at most one vertex in L_2 to the solution.



Thus there is tree of size at most $|OPT|\ (2 \ln \Delta + 1)$ with total profit at least Q



Using the 2-approximation for QST we obtain a |OPT| (4 ln Δ + 2) approximation.



Outline

- Problems
  - Connected dominating set.
  - Generalizations and variants.
- Our Work
  - Summary
  - Algorithm
  - Analysis
- **Future Work**

Future Work

- Our algorithms are tight up to a constant factor. Can we improve the constants
- CDS has good approximation algorithms in the distributed setting. Can we obtain similar algorithms for the partial and budgeted CDS problems ?

THANK YOU FOR LISTENING !!

QUESTIONS?